DIS Superior Conferencing Solutions

A **SHURE** Company

User Manual

SW 6086 Web Service Interface, Version 6

1 Table of Contents

2 Introduction

The purpose of this document is to describe the requirements and information flow for the interface between a SW6000 system and a 3rd party web service that communicates with a Video Streaming system.

Sections changed compared to 'SW 6086 Web Service Interface, Version 5' is marked with <mark>yellow background</mark>.

3 Compatibility

Please refer to the compatibility table on the first page and refer to section '7.6.1 CAA – Setup – Equipment – Web Service'.

4   System applications and abbreviations

Applications in SW6000

CAA – Conference Administration Application

CUA – Conference User Application

CUI – Central Unit Interface

WSI - Web Service Interface – Interface implemented on the SW6000 system

WS - Web Service – Web Service implemented on the server of the Streaming provider

Database – SQL Server

System environment: COM+ application

MSMQ

4.1   Conference Administration Application – CAA

The CAA is an application designed to organize and configure conferences. It's used by Administrators prior to and during an event for creating and maintaining basis conference data such as agendas, delegate information, delegate lists and voting settings.

4.2   Conference User Application – CUA

Used by chairmen, delegates, or other persons attending a conference for viewing the current agenda, delegate information and for starting/stopping conferences and managing microphones and speakers.

4.3   Central Unit Interface – CUI

The CUI application is the Central Unit Interface. As the name indicates it is managing the communication with the CU (Central Unit). Also the CUI is interfacing with all client applications (CUA's, CAA's, CDA's and ECA), The CDA's and ECA are applications used for information sharing and outside the scope of this document.

The client applications will typically receive messages from the CU through the CUI. And finally the CUI is used by client applications to broadcast messages to all other client applications.

4.4   Web Service Interface - WSI

The SW6000 will have an interface through which it communicates with the web service that is implemented on the streaming provider server.

4.5   Web Service

Web services that through methods called by SW6000, communicates with the streaming solution.

4.6   Database

The SW6000 database includes:

Central tables and columns.

Handles multiple languages

"Enumerated types"

5 Solution overview and requirements

An appropriate interface is supplied by SW6000 to communicate with a 3rd party Video Streaming provider.

This chapter will give an overview of this interface, and the following chapters will describe the requirements more detailed.

5.1 Dataflow overview

The SW6000 system will deliver data to the streaming solution. The interface between those two systems is implemented using a web service interface, through a web service running on the streaming solution server. See Figure 1.

CUI broadcast system information originating at the CU, or a client CUA/CAA – with data taken from the SW6000 database.

The WSI then posts the data to the Web Service (WS) on the Streaming solution server. The WS delivers the data sent through the WS to the Streaming solution.

The Video Streaming Solution delivers a controllable webpage, from where it's possible to configure the video streaming. This webpage is embedded in the CAA application for access by an SW6000 administrator.

Handshaking and communication overview is handled between the WSI and WS.

The communication status between the WSI and WS is designed in a way so that minimal impact on the performance of the CUI is achieved.
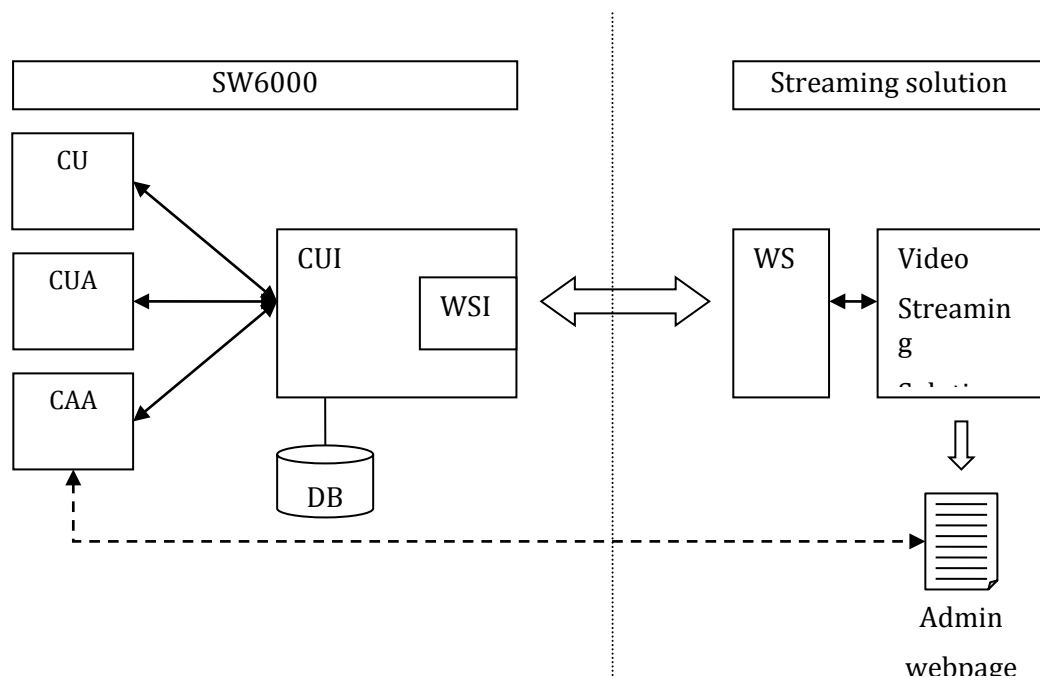
Figure 1

6 Initialization/restart/recovering

The WSI is responsible for keeping the WS updated with correct status of the SW6000 system. During normal operation changes to SW6000 status are transmitted through WS calls.

Communication from both WS and WSI is monitored by the other party in order to detect a failure situation – if the WS stops responding for some reason this will be evident to the WSI due to missing replies on monitoring calls. If a failure occurs – e.g. due to a restart of one of the computers it is ensured that the systems are able to resynchronize full status by going through an initialization sequence.

7 Communication flow

7.1 Web service methods

The web service must support several events from the WSI.

When an event occurs on the SW6000 system, the WSI calls the appropriate Web Service method.

Methods to be implemented in the WS are described in the following subchapters.

Most Methods has xml strings as input parameters. These xml strings are output from datasets on the WSI and it is therefore very useful to read these xml strings into datasets in the WS using a standard xml parser for further data handling.

7.1.1 PrepareMeeting

Input parameters: **strMeetingXML** *[STRING]*

XML format:

```
<newdataset>
    <meeting>
        <system_id>Mødelokale 1<system_id>
        <password>Password1234<password>
        <title>Copenhagen Climate Conference 2009</title>
        <starttime>2009-11-07T12:09:25.7810767+01:00</starttime>
        <description>Discussions about conference schedule</description>
        <public>1</public>
        </publishing>
        <searchtags>UN, United Nations, copenhagen climate conference,
          UNFCCC, Subsidiary Body for Implementation, SBI, SBSTA,
          climate, kyoto</searchtags>
        <categories>Ungdomstinget</categories>
        <category_ids>5</category_ids>
        <web_id>1323</web_id>
    </meeting>
    <timestamp>
        <eventtime>2009-10-27T12:09:25.7810767+01:00</eventtime>
    </timestamp>
</newdataset>
```

Parameter description:

system_id	A unique string for identifying the system the meeting is being held on. I.e. this identifies the meeting room to the WebService making it possible for the WebService to identify multiple meeting rooms using the same WebService.
password	A password string used for identification with the system_id. This is the same password as is sent to the WebService in the VerifyPassword WebService call.
title	Title of the meeting
starttime	Date and time of planned meeting start. Will contain a date without a time part if no start time has been specified for the meeting.
description	Description of the meeting

public	Is meeting to be displayed publicly online (1) or is it private (0).
publishing	Publishing method for meeting recordings. Currently not used.
searchtags	Search keywords to be tagged on meeting recordings.
categories	Names of categories under which to publish meeting recordings. Category names are separated by a semicolon if multiple categories are present.
	To avoid illegal category separation by user-typed semicolons in categories the following character replacements are performed on the individual category names:
	: (colon) => :. (colon period) ; (semicolon)=> :, (colon comma)
	For example the following two category names: - Salen: Evaluering - Belysning; En diskussion would be encoded as: <categories>Salen:. Evaluering;Belysning:, En diskussion</categories>
	The reverse replacements must be made in the receiving end.
category_ids	String of unique identifier integers for the categories named in the "categories" tag. One ID must exist for each category named in the "categories" tag. IDs are separated by a semicolon. The ID for a category is what defines the category – the name is just a humanly readable name related to that ID which is to be used when presented to a user (e.g. if presenting meetings by category in an external application). If an ID is specified and the accompanying category name is different from what has been previously sent for that category ID, the receiving system should update its name for the category to the new name.
web_id	Unique identification of the meeting in the WebService. When creating a new meeting this field is not specified (not present or empty), but if updating data on a meeting already prepared once, this is the ID returned by the PrepareMeeting() WebService function when the meeting was initially prepared.
eventtime	When the event occurred

Return values: A complex value indicating the result of preparing the meeting in the external system

```
<result>

  <status>Success</status>

  <web_id>3214</web_id>

  <message>Some error message – if applicable</message>
```

```
                        </result>
```

	result	Container tag
	status	Status of preparing the meeting. One of the following

- Success. The meeting has been prepared in the external system.

- Failure. The meeting could not be prepared in the external system. The <message> tag contains an error description.

- Duplicate. The meeting has already been prepared. Meeting data has been updated to reflect the new data, so this is also a success indicator.

	web_id	The ID of the conference in the external system. This is what you'll pass as "web_id" argument to calls which require a web_id argument.
	message	An error message. Only valid when <status> is Failure.

Description: Called when a new meeting has been added to make the WebService aware of it before the meeting is referenced in other webservice calls. May also used to update meeting data for a meeting which the webservice has previously been made aware of – in this case a web_id field is contained in the data.

7.1.2   MeetingStart and MeetingStop

Input parameters: strMeetingTitleXML   [STRING]

XML format:

```
<newdataset>
    <meeting>
        <title>Copenhagen Climate Conference 2009</title>
        <import_id>6066</import_id>
        <web_id>1323</web_id>
        <categories>Ungdomstinget</categories>
        <category_ids>5</category_ids>
    </meeting>
    <timestamp>
        <eventtime>2009-10-27T12:09:25.7810767+01:00</eventtime>
    </timestamp>
</newdataset>
```

Parameter description:

	title	Title of the meeting
	import_id	Unique number identifying an imported meeting
	web_id	Unique identification of the meeting in the WebService. This is the ID returned by the PrepareMeeting() WebService function.
	categories	Names of categories assigned to the meeting. See PrepareMeeting call (section 5.1.1) for more information on this tag.

category_ids	String of unique identifier integers for the categories named in the "categories" tag. One ID will exist for each category named in the "categories" tag. See PrepareMeeting call (section 5.1.1) for more information on this tag.
eventtime	When the event occurred

Return values: empty string   [STRING]

Description: Called when meeting/conference starts or stops. Note the dataset may be empty for the MeetingStop call – this is the case when MeetingStop is called as part of the startup sequence and no conference is active. Note the MeetingStart and MeetingStop are always followed by an AgendaUpdate - if the MeetingStart event is called and the meeting has an Agenda the AgendaUpdate call will contain the meeting agenda, if meeting has no agenda or MeetingStop is called the subsequent AgendaUpdate call will contain an empty agenda.

7.1.3   SpeakerChange

Input parameters: strSpeakerListXML   [STRING]

XML format:

```
<newdataset>
  <seat>
    <title>Minister</title>
    <name>Anne Baastrup</name>
    <usertype>Delegate</usertype>
    <usertable1 />
    <usertable2 />
    <usertable3 />
    <usertable4 />
    <group>Socialist Group</group>
    <groupabbreviation>SG</groupabbreviation>
    <votegroup />
    <seatnumber>5</seatnumber>
    <delegate_id>73</delegate_id>
    <import_id>100589</import_id>
  </seat>
   .
   .
   .
  <seat>
    <title />
    <name>Anita Knakkergaard</name>
    <usertype>Delegate</usertype>
    <usertable1 />
    <usertable2 />
    <usertable3 />
    <usertable4 />
    <group>Danish Peoples Group</group>
    <groupabbreviation>DPG</groupabbreviation>
    <votegroup />
    <seatnumber>4</seatnumber>
    <delegate_id>132</delegate_id>
    <import_id>100576</import_id>
  </seat>
  <timestamp>
    <eventtime>2009-10-27T13:52:54.1023068+01:00</eventtime>
  </timestamp>
</newdataset>
```

Parameter description:

title	Title of the delegate
name	Display name of the delegate.
usertype	privileges of the delegate (as setup in CAA)
usertable1	as setup in CAA
usertable2	as setup in CAA
usertable3	as setup in CAA
usertable4	as setup in CAA
group	as setup in CAA
groupabbreviation	as setup in CAA
votegroup	as setup in CAA
seatnumber	identifying microphone position
delegate_id	unique number identifying delegate [note two delegates may have identical names]
import_id	unique number identifying delegate imported (TingDok Id)
eventtime	When the event occurred

Return values: empty string   [STRING]

Description: Called when speaker changes
XML will contain Seat information for all current speakers. XML will only contain eventtime if there are no speakers.

7.1.4 VotingStart

Input parameters: strSubjectNameXML   [STRING]

                      XML format:

```
<newdataset>
  <currentsubject>
     <level>1</level>
     <name>Global Warming</name>
     <description />
     <shortdescription />
     <agenda_id>2653</agenda_id>
     <import_id>32332</import_id>
     <itemnumber>1</itemnumber>
  </currentsubject>
  <currentsubject>
     <level>2</level>
     <name>Ice Caps</name>
     <description />
     <shortdescription />
     <agenda_id>2654</agenda_id>
     <import_id />
     <itemnumber>1.1</itemnumber>
  </currentsubject>
  <currentsubject>
     <level>3</level>
     <name>Rising Water Levels</name>
     <description />
     <shortdescription />
     <agenda_id>2655</agenda_id>
     <import_id />
     <itemnumber />
  </currentsubject>
  <timestamp>
     <eventtime>2009-10-27T14:07:44.0956233+01:00</eventtime>
  </timestamp>
</newdataset>
```

Parameter description:

level	Identifies the hierarchy of subject/parent subjects. Level 1 is main, and 2.. are sub items.
name	Name of the current subject as well as name of parent subjects to the current subject
description	agenda item description
shortdescription	agenda item short text description
agenda_id	Unique number for agenda item – SW 6000 autogenerated
Import_id	Unique number identifying delegate imported (TingDok Id)
itemnumber	Item Number text as input in agenda from CAA.
Eventtime	When the event occurred

Return values: empty string   [STRING]

Description: Called when voting starts. Part of the system update procedure. In a WS restart sequence, it is called if a voting session is in progress.

If voting session is started without any subject selected, the dataset will be empty.

7.1.5   **VotingStop**

Input parameters: strVotingResultsXML   [STRING]

XML format (Complete content of Session dataset):

```
<newdataset>
  <session>
     <session_id>762</session_id>
     <version>1</version>
     <date>2009-10-28T08:46:50.6205579+01:00</date>
     <passedresult>1</passedresult>
     <passedresulttext>Passed</passedresulttext>
     <result1>1</result1>
     <result2>0</result2>
     <result3>0</result3>
     <result4>0</result4>
     <result5>1</result5>
     <result6>0</result6>
     <result7>0</result7>
     <result8>0</result8>
     <result9>0</result9>
     <result1label>Yes</result1label>
     <result2label>No</result2label>
     <result3label>Abstain</result3label>
     <result4label />
     <result5label>Total</result5label>
     <result6label />
     <result7label />
     <result8label />
     <result9label />
     <conclusion>'Conclusion text    '</conclusion>
  </session>
  <timestamp>
     <eventtime>2009-10-28T09:45:51.2541477+01:00</eventtime>
  </timestamp>
  <currentsubject>
     <level>1</level>
     <name>Global Warming</name>
     <description />
     <shortdescription />
     <agenda_id>2653</agenda_id>
     <import_id />
     <itemnumber>1</itemnumber>
  </currentsubject>
  <currentsubject>
     <level>2</level>
     <name>Ice Caps</name>
     <description />
     <shortdescription />
     <agenda_id>2654</agenda_id>
     <import_id />
     <itemnumber>1.1</itemnumber>
  </currentsubject>
  <currentsubject>
     <level>3</level>
     <name>Rising Water Levels</name>
     <description />
     <shortdescription />
     <agenda_id>2655</agenda_id>
     <import_id />
     <itemnumber />
  </currentsubject>
</newdataset>
```

Parameter description:

session_id	Internal identifier for voting session
version	Sequence number identifying session – if revotes are held on a subject the sequence number will be greater than 1 on revotes
date	Voting date
passedresult	1: Vote Passed, 0: Vote Not Passed
passedresulttext	Text to display for vote result
result1 to result9	Calculated results of voting session.
result1label to result9	label Labels associated with calculated voting results.
conclusion	Text overview of voting result
eventime	When the event occurred
level	Identifies the hierarchy of subject/parent subjects. Level 1 is main, and 2.. are sub items.
name	Name of the current subject as well as name of parent subjects to the current subject
description	agenda item description
shortdescription	agenda item short text description
agenda_id	Unique number for agenda item – SW 6000 autogenerated
Import_id	Unique number identifying delegate imported (TingDok Id)
itemnumber	Item Number text as input in agenda from CAA.

Return values: empty string [STRING]

Description: Called when voting stops or the voting session is cancelled. VotingStop is part of the system update procedure. If WS fails, VotingStop is called when connection is re-established if no voting session is in progress and the current subject has a voting result attached with the corresponding voting result. On startup or WS connection re-established when currently no meeting is active or a meeting is active but no subject is current or a subject is current with no voting result the VotingResult dataset will contain empty tags.

When a voting session completes VotingStop is called with VotingResult dataset containing the result of the voting session. If a voting session is cancelled, VotingStop will be called with VotingResult dataset containing all empty tags.

7.1.6   VotingHide

Input parameters: No input

Return values: empty string [STRING]

Description: Called at a specified delay after voting stop calls. To be used by WebService to remove the voting results Webpage from being the active display. The delay interval is retrieved from the database.

7.1.7   IndividualVotingResult

Input parameters: strIndividualVotingResultXML [STRING]

XML format (Complete content of Session dataset):

```
<newdataset>
  <currentsubject>
     <level>1</level>
     <name>Global Warming</name>
     <description />
     <shortdescription />
     <agenda_id>2653</agenda_id>
     <import_id />
     <itemnumber>1</itemnumber>
  </currentsubject>
  <currentsubject>
     <level>2</level>
     <name>Ice Caps</name>
     <description />
     <shortdescription />
     <agenda_id>2654</agenda_id>
     <import_id />
     <itemnumber>1.1</itemnumber>
  </currentsubject>
  <currentsubject>
     <level>3</level>
     <name>Rising Water Levels</name>
     <description />
     <shortdescription />
     <agenda_id>2655</agenda_id>
     <import_id />
     <itemnumber />
  </currentsubject>
  <result>
     <session_id>762</session_id>
     <name>Anne Grete Holmsgaard</name>
     <group>Socialistisk Folkeparti</group>
     <groupabbreviation>SF</groupabbreviation>
     <votegroup>SF</votegroup>
     <delegate_id>109</delegate_id>
     <import_id>100691</import_id>
     <voteweight>1</voteweight>
     <vote>1</vote>
     <votetext>For</votetext>
  </result>
  <timestamp>
     <eventtime>2009-10-28T09:45:51.2541477+01:00</eventtime>
  </timestamp>
</newdataset>
```

Parameter description:

level	Identifies the hierarchy of subject/parent subjects. Level 1 is main, and 2.. are sub items.
name	Name of the current subject as well as name of parent subjects to the current subject
name	Agenda item title
description	Agenda item description
shortdescription	Agenda item short text description
import_id	unique number identifying delegate imported (TingDok Id)
agenda_id	unique number for agenda item SW 6000 auto generated

	itemnumber	Item Number text as input in agenda from CAA
	session_id	Unique identifier for voting session
	name	First/Last name of the delegate
	group	as setup in CAA
	groupabbreviation	as setup in CAA
	votegroup	as setup in CAA
	delegate_id	unique number identifying delegate [note two delegates may have identical names]
	voteweight	as setup in CAA
	vote	choice 1-5 according to button pressed
	votetext	choice according to voting configuration in use (button text).

Description: Individual voting results is called on the voting session close event. Note individual voting results are only for voting sessions held during a conference if delegates are logged in or on a prepared delegate list. Individual voting results are not sent for voting sessions held as secret voting.

7.1.8 StatusUpdateStart and StatusUpdateDone

Input parameters: No input

Return values: empty string   [STRING]

Description: Used as wrappers for system initialization after reboot/restart. See 6.4

When Web Service receives a StatusUpdateStart any current status should be updated accordingly – e.g. after a CUI restart with a running Web Service/Streaming solution.

7.1.9 StreamingStart

Input parameters: strStreamingModeXML   [STRING]

XML format:

```
<newdataset>
  <streamingmode>
    <mode>Broadcast</mode>
  </streamingmode>
  <timestamp>
     <eventtime>2009-10-27T14:07:44.0956233+01:00</eventtime>
  </timestamp>
</newdataset>
```

Parameter description:

	mode	Streaming mode. Values: Values: Broadcast, Archive or BroadcastArchive

Return values: empty string   [STRING]

Description: Called by user from within the CUA

7.1.10   **StreamingStop**

Input parameters: strStreamingStopXML

                            XML format:

```
<newdataset>
  <timestamp>
     <eventtime>2009-10-27T14:07:44.0956233+01:00</eventtime>
  </timestamp>
</newdataset>
```

Return values: empty string   [STRING]

Description: Called by user from within the CUA

7.1.11   **AgendaUpdate**

Input parameters: strAgendaXML   [STRING]

                            XML Format (Complete content of Agenda dataset):

```
<newdataset>
  <agenda>
    <level>1</level>
    <sortorder>1</sortorder>
    <iscurrent>1</iscurrent>
    <name>1. Welcome</name>
    <description>Text description 1</description>
    <shortdescription>Short text description 1</shortdescription>
    <agenda_id>65</agenda_id>
    <import_id />
    <itemnumber />
  </agenda>
  <agenda>
    <level>2</level>
    <sortorder>2</sortorder>
    <iscurrent>2</iscurrent>
    <name>1.1. Introduction</name>
    <description />
    <shortdescription />
    <agenda_id>66</agenda_id>
    <import_id />
    <itemnumber />
  </agenda>
  <agendalink>
    <agenda_id>66</agenda_id>
    <sortorder>1</sortorder>
    <linknumber />
    <linkname>SAMPLE.PDF</linkname>
    <linkurl>X:\SAMPLE.PDF</linkurl>
  </agendalink>
  <agenda_event>
    <type>AgendaUpdate</type>
  </agenda_event>
  <timestamp>
    <eventtime>2012-09-17T14:20:07.8621441+02:00</eventtime>
  </timestamp>
</newdataset>
```

Parameter description:

	level	Indent level
	sortorder	Display order of the items in the xml file. (item number)
	iscurrent	If all items are 0, no item is current. If an item is 2, it's the current item, parent items to the current item will be 1 – facilitating identification.
	name	Agenda item displayed text – field is formatted according to standard. If no text is entered for an agenda item but a speech type/speaker has been entered these will display in the text.
	description	Agenda item description
	shortdescription	short text description
	import_id	Id originating from import
	agenda_id	Unique id for agenda item SW 6000 autogenerated
	itemnumber	Item Number text as input in agenda from CAA
	linknumber	The link number text for an agenda link
	linkname	The link name for an agenda link
	linkurl	The link target URL for an agenda link. The target can be any URI – typically a file or a web reference (http, ftp etc.).
	type	Identifies if the agenda update is due to either a change in subject (AgendaSubject) or a change in agendacontent (AgendaUpdate).
	eventtime	When the event occurred
Return values:	empty string	[STRING]

Description: Called when Agenda content changes, at start and stop of meeting and at startup with event type **AgendaUpdate**. Also called when an agenda item is set active or no agenda item active with event type **AgendaSubject**. Note that if any of the fields are not available in the requested language, these fields will be transmitted in the default language, and if not available in the default language an empty field is transmitted.

7.1.12   DelegateDetails

Input parameters: strDelegateInfoXML   [STRING]

XML format:

```
<newdataset>
  <delegate>
     <title>Statsministeren</title>
     <first_name>Anders Fogh</first_name>
     <last_name>Rasmussen</last_name>
     <show_name>Statsministeren</show_name>
     <usertype>Minister</usertype>
     <usertable1 />
     <usertable2 />
     <usertable3 />
     <usertable4 />
     <group />
     <votegroup />
     <background>additional information</background>
     <delegate_id>205</delegate_id>
     <import_id>100735</import_id>
     <photos>/9j/4AAQSkZJRgABA …
               …
… ABQAUAFABQAUAFABQAUAFABQAUAFABQAUAFID//ZAA==</photos>
  </delegate>
  <timestamp>
     <eventtime>2009-10-28T10:25:32.4889077+01:00</eventtime>
  </timestamp>
</newdataset>
```

Parameter description:

title	Delegate title
first_name	First name of the Delegate
last_name	Last name of the Delegate
show_name	Display name according to user type
usertype	Type of user. E.g. Delegate or Chairman
usertable1 to 4	Four user-defined fields, available for entering various text information. E.g. Country or department.
group	As User table, normally used for Party
votegroup	User defined field – used to differentiate delegates when using voting formulas
background	User defined field – available for entering text describing details of delegate – can be up to 1000 char, and multiline
delegate_id	Id of Delegate
import_id	Id originating from import
photos	serialized byte array of delegate picture
eventtime	When the event occurred

Note that if any of the fields are not available in the requested language, these fields will be transmitted in the default language, and if not available in the default language an empty field is transmitted – see example.

Description: The DelegateDetails function has a parameter which includes the delegate picture and this parameter contains a large amount of data if pictures have been assigned to delegates. To reduce communication of large amounts of data to the

webservice the DelegateDetails may be omitted depending on the setup in the CAA, see 06.6.1.

DelegateDetails are sent within the StatusUpdate process (see 6.1.5) if a meeting is in progress and delegates participating in the meeting are identified – (on a delegate list). If a meeting with login mode 'login using code' is in progress only delegates logged in will have details sent. On subsequent login's from other delegates the details of these delegates will be transmitted as the delegates perform login.

Note that the DelegateDetails call has always been made prior to DelegateLogin call – thus details are always available when the DelegateLogin call is made.

7.1.13 DelegateLogin

Input parameters: strLoginString   [STRING]

XML format:

```
<newdataset>
  <login>
    <seat>30</seat>
    <usertype>Minister</usertype>
    <show_name>Statsministeren</show_name>
    <delegate_id>205</delegate_id>
    <import_id>100735</import_id>
  </login>
  <timestamp>
    <eventtime>2009-11-13T14:00:49.4525549+01:00</eventtime>
  </timestamp>
</newdataset>
```

Parameter description:

seat	Identification of seat where login is performed
usertype	Type of user. E.g Delegate or Chairman
show_name	Display name according to usertype
delegate_id	Id of Delegate
import_id	Id originating from import
eventtime	When the event occurred

Description: The DelegateLogin call is made whenever a delegate is logged in to a seat. For mode "Prepared Delegate Seat" a "DelegateLogin" is sent for all delegates on the list on conference start. For conference modes where login is required, the "DelegateLogin" call is made when an actual login is performed. Note that details for the delegates are made available in the "DelegateDetails" calls. A delegate will only be logged in once for mode "Prepared Delegate Seat", while in modes using login e.g. "Login Using Code" the delegate may be logged in several times at different locations, and possibly with different user types. For mode "Code and List" and "Code on list" the "DelegateDetails" call is only made once – but as the delegate may be logged in several times with a new usertype it's important that the usertype and show_name settings are updated on a new login. Pictures are not updated and a performance gain is achieved by not sending delegate details for each login when using these modes. A single call may contain multiple login entries.

7.1.14   DelegateLogout

Input parameters: strLogoutString   [STRING]

XML format:

```
<newdataset>
  <logout>
    <seat>30</seat>
    <delegate_id>205</delegate_id>
    <import_id>100735</import_id>
  </logout>
  <timestamp>
    <eventtime>2009-11-13T14:00:49.4525549+01:00</eventtime>
  </timestamp>
</newdataset>
```

Parameter description:

seat	Identification of seat where login is performed
delegate_id	Id of Delegate
import_id	Id originating from import
eventtime	When the event occurred

Description: The DelegateLogout call is made whenever a delegate is logged out from a seat. For conference modes where login is required, the "DelegateLogout" call is made when an actual logout is performed. When a conference requiring login is stopped all delegates are logged out, this is also true for conference mode "PreparedDelegateSeat". A single call may contain multiple logout entries. Stopping a conference is equivalent of performing logout for all delegates.

7.1.15   AlertStart

Input parameters: strAlertString   [STRING]

XML format:

```
<newdataset>
  <alert>
    <text>A</text>
  </alert>
  <timestamp>
    <eventtime>2009-11-13T14:00:49.4525549+01:00</eventtime>
  </timestamp>
</newdataset>
```

Parameter description:

alert	Alert text as setup for the conference
eventtime	When the event occurred

Return values: empty string   [STRING]

Parameter description:

text

Description: This alert originates from a CUA user with appropriate user type who activates the alert. Note that alerts are automatically deactivated when a voting session is

stopped. The input parameter alert string is the string to display to users identifying this alert.

7.1.16   AlertStop

Input parameters: strAlertStop   [STRING]

XML format:

```
<newdataset>
  <timestamp>
    <eventtime>2009-11-13T14:06:47.9442398+01:00</eventtime>
  </timestamp>
</newdataset>
```

Parameter description:

eventtime	When the event occurred

Return values: empty string   [STRING]

Description: Removes a previously issued alert.

7.1.17   WebServiceStatus

Input parameters: No input

Return values: WebServiceStatus   [STRING]

See Description

Description: The Web Service can return the current running state. The different states are described below:

Unavailable	Streaming system not available but Web Service is running
Needstatus	Streaming system is just started (could be after a shutdown/reboot). Indicates that the WS needs status/information from the SW6000 WSI
Ready	Status is updated but no streaming has started (not running)
Broadcasting	Broadcasting is running (set after StreamingStart is called)
Archiving	Archiving
BroadcastingArchiving	Broadcasting and Archiving are running

If status is Ready, Broadcasting, Archiving or BroadcastingArchiving, the Streaming Start/Stop button on the CUA is enabled. In these states status changes in SW6000 are transmitted to the WS (e.g. AgendaUpdate, SpeakerChange …)

If status is Needstatus, password verification process is executed before any other action will work. See 0.

The WebServiceStatus call is issued repeatedly at a 10 second interval to monitor the WebService.

7.1.18   RequestDisplayLanguage

Input parameters: No input

Return values: LanguageId   [INTEGER]

SW6000 supports multiple languages; the streaming solution should be able to be set up for any of the supported languages in the SW6000 database (as configured by the CAA).

E.g 1033 for English (US), 1030 for Danish and 1053 for Swedish.

If a language id is returned which is not supported in the Database for the current meeting then all strings sent to the WS will be in the default language as defined in the SW6000 database.

7.1.19   RequestStreamingSupport

Input parameters: No input

Return values: StreamingSupport   [STRING]

     Values:

     Broadcast

     Archiving

     BroadcastArchiving

The StreamingSupport value returned tells the DIS system what configuration the streaming solution supports. Just a direct broadcast (Broadcast), just archiving (Archiving) or simultaneous direct broadcast and archiving (BroadcastArchiving).

7.1.20   RequestVoteHideDelay

Input parameters: No input

Return value: VoteHideDelay   [STRING] (is convertible to integer)

The VoteHideDelay vaue returned tells the DIS system the delay in seconds between a VotingStop call and the subsequent VotingHide call. Default value is 3 seconds.

7.1.21   VerifyPassword

Input parameters: strPassword   [STRING] WebServicePassword

Return value: strPassword   [STRING] StreamingPassword

Two passwords are used to establish communication – WebServicePassword is used in the call from Web Service Interface, and Streaming password should be returned in the reply from the WebService. These passwords are setup in the CAA application and should match in order to initiate communication.

7.2   Handling Agenda changes

When changes are made in the Agenda, this is sent to the Streaming solution

A change to the Agenda may include insertion or deletion of agenda items as well as changes to the current active subject, including the change to set no agenda item active. Changes also include sending an empty agenda e.g. on meeting stop. To differentiate between agenda updates involving agenda content and agenda updates changes to the current agenda item the agenda event type is available. An agenda is a hierarchical structure where sub items should be interpreted in relation to their parent items. To facilitate identification of this parent child relationship for the active agenda item, parent agenda items that act as parent to the active agenda item have isCurrent set to 1, and the active agenda item has isCurrent set to 2.

WSI   WS

AgendaUpdate
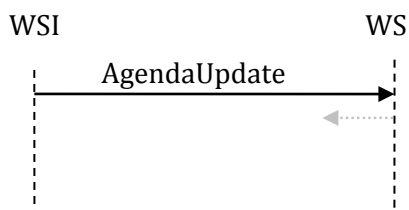
7.3   Handling Status request

The WSI must query WS for status every 10 seconds.

If status differs from "Ready", appropriate action is taken. See 6.1.14

WSI   WS

WebServiceStatus

Status: Ready
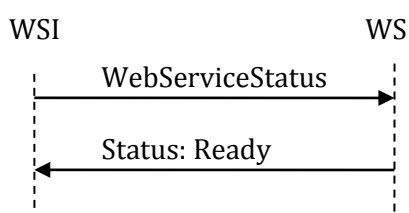
7.4   Handling web service and SW6000 reboot/restart/breakdown

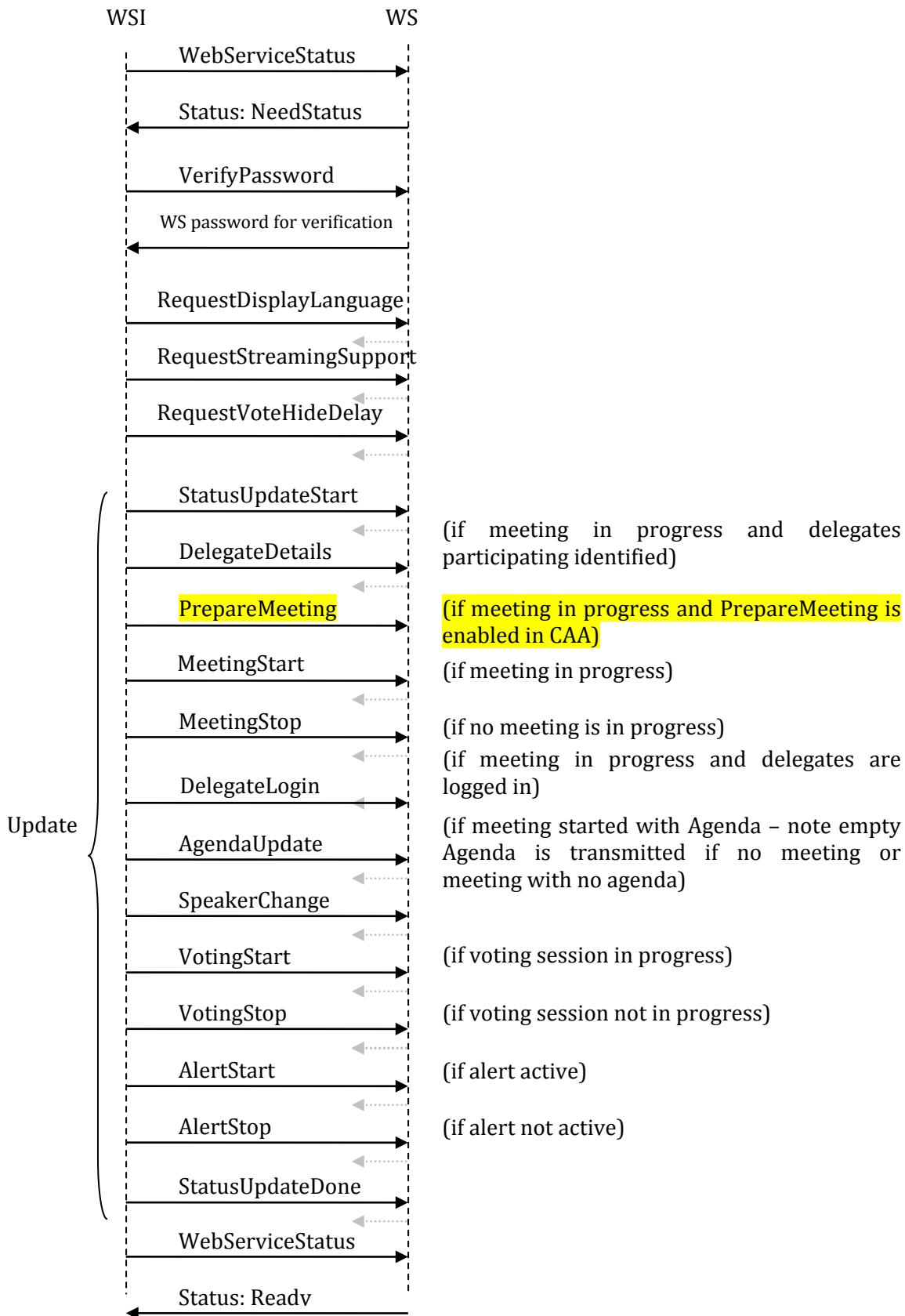The system will handle reboot/restart/breakdown and turn into actual state by itself.

If CUI is restarted on a running Web Service the Web Service will return e.g. Ready or Broadcasting, Archiving or BroadcastingArchiving. Still status is updated according to flow depicted.

Below is an event diagram of the flow if this happens

7.4.1   Starting up Web service while Web Service Interface (CUI) is running

7.4.2　Starting up Web Service Interface (CUI) while Web Service is running

```
        WSI                    WS
         │  WebServiceStatus    │
         ├─────────────────────►│
         │                      │
         │  Status: NeedStatus  │
         │◄─────────────────────┤
         │                      │
         │  VerifyPassword      │
         ├─────────────────────►│
         │                      │
         │ WS password for verification
         │◄─────────────────────┤
         │                      │
         │  RequestDisplayLanguage
         ├─────────────────────►│
         │◄╴╴╴╴╴╴╴╴╴╴╴╴╴╴╴╴╴╴╴╴╴│
         │  RequestStreamingSupport
         ├─────────────────────►│
         │◄╴╴╴╴╴╴╴╴╴╴╴╴╴╴╴╴╴╴╴╴╴│
         │  RequestVoteHideDelay│
         ├─────────────────────►│
         │◄╴╴╴╴╴╴╴╴╴╴╴╴╴╴╴╴╴╴╴╴╴│
```

	StatusUpdateStart	
	DelegateDetails	(if meeting in progress and delegates participating identified)
	PrepareMeeting	(if meeting in progress and PrepareMeeting is enabled in CAA)
Update	MeetingStart	(if meeting in progress)
	MeetingStop	(if no meeting is in progress)
	DelegateLogin	(if meeting in progress and delegates are logged in)
	AgendaUpdate	(if meeting started with Agenda – note empty Agenda is transmitted if no meeting or meeting with no agenda)
	SpeakerChange	
	VotingStart	(if voting session in progress)
	VotingStop	(if voting session not in progress)
	AlertStart	(if alert active)
	AlertStop	(if alert not active)
	StatusUpdateDone	

```
         │  WebServiceStatus    │
         ├─────────────────────►│
         │◄╴╴╴╴╴╴╴╴╴╴╴╴╴╴╴╴╴╴╴╴╴│
         │  Status: Ready       │
         │◄─────────────────────┤
```

Handling no response/unavailable status from web service

WSI will not send any events to the WS as long as no response or Unavailable status received from WS

On the CUA radio buttons for selecting Streaming mode (Broadcast/Archiving/BroadcastArchiving) is grayed out.

7.5   Handling Streaming Start/Stop

Error! Reference source not found. shows the communication flow at Streaming start/stop
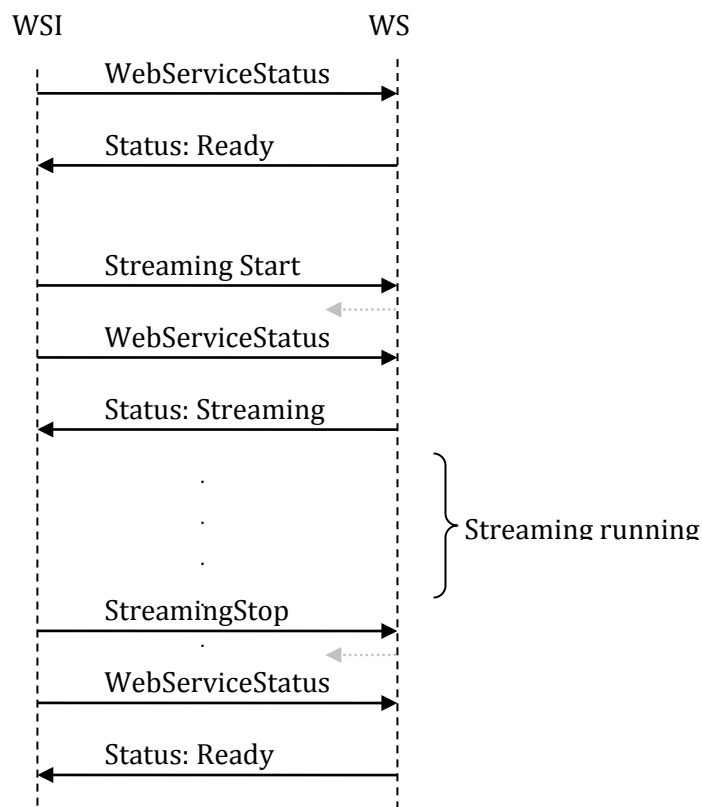
8   Setup of SW6000

8.1   CAA – Setup – Equipment – Web Service

Using the CAA application, the following parameters shall be setup for using Web Service calls:

- Web Service Address
- Web Service Administrator Address
- Passwords for connection between WSI and WS
- Checkboxes to select, which additional information to be send to the web service. This is also used for making the present version of the web service compatibility with previous version of SW 6000.

When checked the information selected is sent to the Web Service. Only information on active delegates is sent.

The WSI will send a password for verification in the WS. Password to be specified in the CAA.
The WS will send a password for verification in the WSI. A verification password is to be specified in the CAA.

8.2   CAA – Embedded Streaming administration page

The Streaming provider must generate a webpage to be embedded in a window "Streaming Configuration" in the CAA application in the Setup -> Configuration section. Size must fit current layout. If more space is required, tabs are used.

Address for this webpage to be specified on the "Streaming" window. See 6.6.1

The webpage should be a minor administration interface that can be managed from the CAA and a status check showing what's going on. In this way the user doesn't have to run around between several computers to view Streaming status, except for trouble shooting.

Following information is implemented on the webpage:

1. Start/stop broadcast (streaming)
2. Start/stop archiving (this setting can be set in the administration interface in the Streaming software so the recording will start automatically when using start Broadcast)
3. Encoder status – showing the status of the encoder
4. Broadcast check (showing the output media stream so it is shown in a player in the CAA. This shows the user that something is going out from the encoder)
5. Edit and publish on demand

### 8.2.1   Changing streaming mode

Changing between Broadcast/Archiving/BroadcastArchiving and displaying current setting is possible in the Streaming Administrator interface - embedded as a webpage in the CAA

Return values for selected streaming mode:

Streaming mode: Broadcast   Return value: Broadcasting

Streaming mode: Archiving   Return value: Archiving

Streaming mode: BroadcastArchiving   Return value: BroadcastArchiving

8.3   CAA – User permissions

Streaming Control is included in the User Type Functionality settings. On the User Type window, add a Checkbox for "Streaming Control". When checked, user can use the Streaming functionality on the CUA.
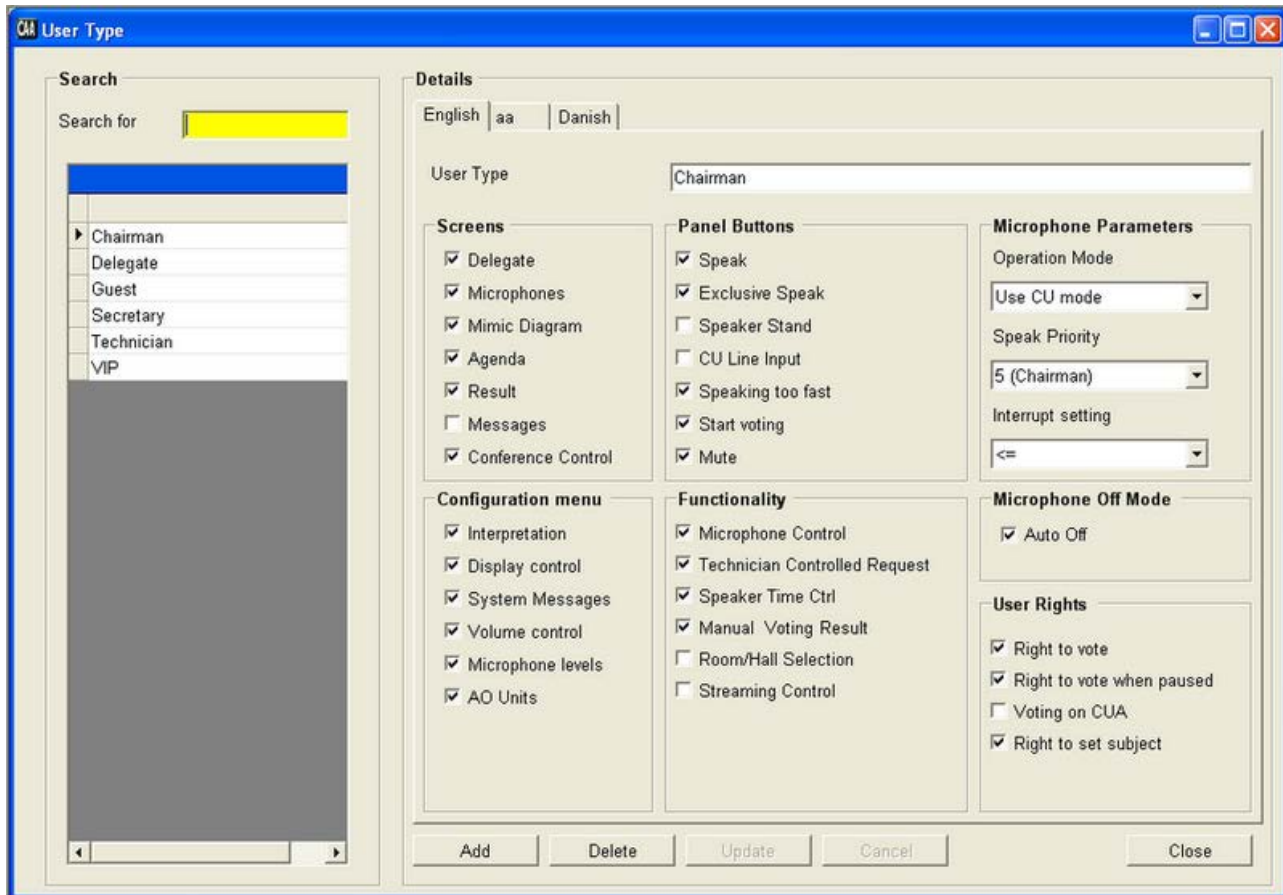
8.4   CUA/CAA – Starting and Stopping streaming and archiving

2 buttons are implemented. One for Starting the Broadcasting/Streaming and one for starting the Archiving/Recording.

Only buttons where the feature is supported by the Streaming solution (answered by the RequenstStreamingSupport) is visible.

Buttons will change color from default to red when streaming/archiving is running. Button only enabled if Web Service is available else locked.

The following restrictions apply:

Archiving is only possible when streaming (broadcast) is active.

Archiving is only possible when a conference is started.

Additionally if the user selects to stop a conference (from CAA or CUA), then a stop archiving message is sent over the webservice interface.
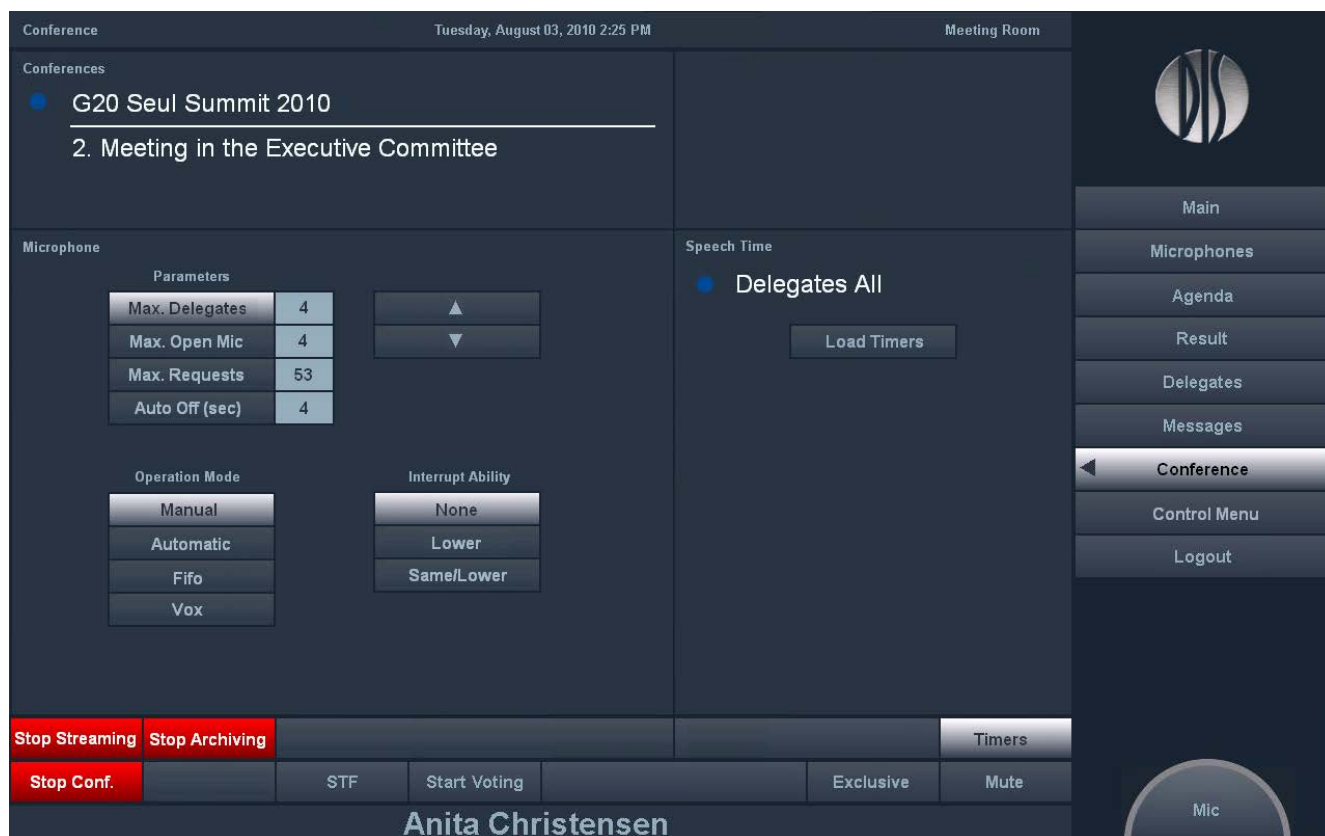
8.5 CUI – Communication Status display

For debugging purpose, the CUI display contains a window showing the feedback from the WSI.

When sending pictures using the DelegateDetails method the picture is included in the XML format. The picture data must not be send back the to CUI window, but be replaced with a <<picture>> tag instead.

9   Security issues

The connection between the Web Service and SW6000 is secured by passwords. See 6.6.1

SHURE ®

LEGENDARY
PERFORMANCE™

Shure Incorporated
5800 West Touhy Avenue
Niles, IL 60714-4608
USA
Tel. +1 847 600 2000
Fax + 1 847 600 1212 (USA)
Fax + 1 847 600 6446
info@shure.com

Danish Interpretation Systems A/S
Vestre Teglgade 12
DK-2450 Copenhagen SV
Denmark
Tel. +45 33 85 40 40
Fax +45 33 85 40 41
salessupport@dis.cc

DIS Middle East
LOB 15; Office 221
Jebel-Ali Free Zone
P.O. Box 17343
UAE
Tel. +971 4 88 71 915
Fax +971 4 88 71 897
dubai@dis.cc

DIS Asia Pacific
1497 Soi Latphrao 94
(Town In Town 3/2)
Latphrao Rd. Plubpla
Wangtonglang
Bangkok 10310
Thailand
Tel. +662 559 3420
Fax +662 559 3424
Bangkok@dis.cc

www.shure.com   @2013 Shure Incorporated   **www.dis.cc**